ARL 72-0014
JANUARY 1972

Aerospace Research Laboratories

CROUT ALGORITHM WITH ACCUMULATED INNER PRODUCT

NAI-KUAN TSAO

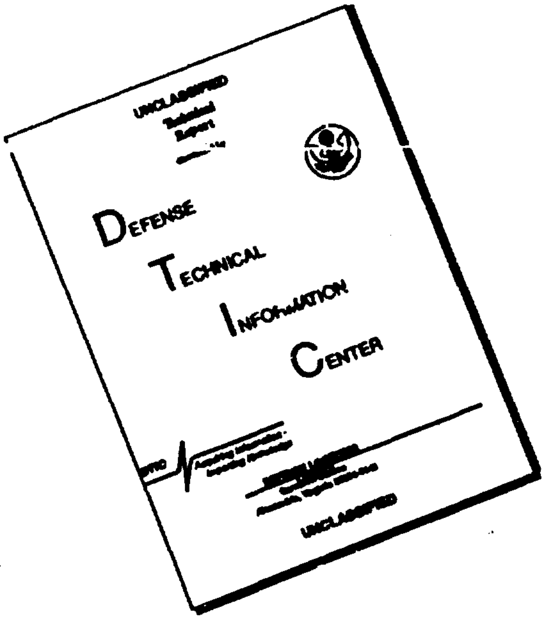
APPLIED MATHEMATICS RESEARCH LABORATORY

PROJECT NO. 7071

Approved for public release; distribution unlimited.

AIR FORCE SYSTEMS COMMAND
United States Air Force

DISCLAIMER NOTICE



UNCLASSIFIED

Technical Report

DEFENSE TECHNICAL INFORMATION CENTER

DTIC

UNCLASSIFIED

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by *implication or otherwise* as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Agencies of the Department of Defense, qualified contractors and other government agencies may obtain copies from the

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

This document has been released to the

CLEARINGHOUSE
U. S. Department of Commerce
Springfield, Virginia 22151

for sale to the public.

Copies of ARL Technical Documentary Reports should not be returned to Aerospace Research Laboratories unless return is required by security considerations, contractual obligations or notices on a specified document.

AIR FORCE: 19-4-72/100

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Aerospace Research Laboratories Applied Mathematics Research Laboratory Wright-Patterson AFB, Ohio 45433	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Crout Algorithm with Accumulated Inner Product

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (First name, middle initial, last name)

Nai-Kuan Tsao

6. REPORT DATE January 1972	7a. TOTAL NO. OF PAGES 21	7b. NO. OF REFS 5

8a. CONTRACT OR GRANT NO. In-house Research b. PROJECT NO. 7071-00-14 c. DoD Element 61102F d. DoD Subelement 681304	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) ARL 72-0014

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES Tech Other	12. SPONSORING MILITARY ACTIVITY Aerospace Research Laboratories (LB) Wright-Patterson AFB, Ohio 45433

13. ABSTRACT

A posteriori forward error analysis is applied to the Crout algorithm with inner product accumulation in solving system of linear algebraic equations of the type $Ax = b$. By attributing the generated round-off errors properly to the matrices A and b, it is shown, under certain reasonable assumptions, that the computed x satisfies a new perturbed system such that $(A + \delta A)x = b + \delta b$ and the upper bounds for δA and δb in infinite norm are shown to be proportional to n, the system order. This is an improvement over the results where the inner products are not accumulated.

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Crout algorithm						
Inner product						
Linear equations						
floating-point arithmetic						
Matrix decomposition						
Error analysis						

CROUT ALGORITHM WITH ACCUMULATED INNER PRODUCT

NAI-KUAN TSAO

APPLIED MATHEMATICS RESEARCH LABORATORY

JANUARY 1972

PROJECT 7071

AEROSPACE RESEARCH LABORATORIES
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

FOREWORD

This research was accomplished while the author held a National Research Council Postdoctoral Resident Research Associateship supported by the Applied Mathematics Research Laboratory of the Aerospace Research Laboratories. The author wishes to thank Dr. Paul J. Nikolai for his encouragements and comments during the preparation of this paper. Thanks are also due to Mrs. Barbara Geiger for a carefully typed manuscript.

ABSTRACT

A posteriori forward error analysis is applied to the Crout algorithm with inner product accumulation in solving system of linear algebraic equations of the type $Ax = b$. By attributing the generated round-off errors properly to the matrices A and b, it is shown, under certain reasonable assumptions, that the computed x satisfies a new perturbed system such that $(A + \delta A)x = b + \delta b$ and the upper bounds for δA and δb in infinite norm are shown to be proportional to n, the system order. This is an improvement over the results where the inner products are not accumulated.

TABLE OF CONTENTS

1. Introduction.

In solving system of linear equations of the type $Ax = b$ where A is a non-singular n-tn order matrix and b is an n-vector, the Gaussian elimination method of decomposing A into a product LU of a lower triangular matrix L and an upper triangular matrix U probably is the most generally used algorithm because its economy in the number of arithmetic operations required and the numerical stability of the solution. For normalized floating-point computations with t-bits allocated to the mantissa of a floating-point number, we have

$$(1 + \delta)\ fl(x*y) = x*y, \qquad |\delta| \leq 2^{-t} = u \tag{1.1}$$

where * is any of the operators +, -, ×, /. Under the condition of (1.1), it is shown [1] that the computed x using the Gaussian algorithm satisfies a new perturbed system such that

$$(A + \delta A)x = b + \delta b \tag{1.2}$$

and

$$\begin{aligned} ||\delta A||_\infty &\leq (n^2 - 1)\sigma u, \\ ||\delta p||_\infty &\leq (n^2 + n - 1 + n\sigma)\rho u \end{aligned} \tag{1.3}$$

where σ and ρ are some constants obtainable after the computation. Equation (1.1) implies that for the given two numbers x and y with t-bits mantissae, $fl(x*y)$ is the correctly rounded result of the floating operation *. It is shown that the operations + and - are ill-conditioned in the sense of

Rice [2] if $|x \pm y|$ is very small. In othe words, there is a loss of significance if $|x \pm y|$ is considerably smaller than $|x| + |y|$. However, the relative condition of $x \times y$ or $x \div y$ is a constant 2. Hence for ill-conditioned systems, the loss of significance due to additive operations in the early stages of computation might lead to unacceptable final solutions. Thus one remedy for such systems is the use of higher-precision arithmetics at the expense of more computing time and memory space. Another alternative is to add or subtract in double-precision whereas multiplication and division could still be done in single-precision. Furthermore, the result of a single-precision multiplication can easily be retained in double-precision and used later. This is extremely helpful in the computation of inner products. This type of computation can thus be called "accumulated inner product" arithmetic.

The Crout variation of the Gaussian elimination methods is essentially a sequence of inner product computations. Hence the use of accumulated inner product should improve the accuracy in the final solutions. In this paper we will carry out the a posteriori forward error analysis [3] of the Crout algorithm with accumulated inner product. The results show that the computed solution satisfies a perturbed system similar to (1.2) with bounds for the perturbations proportional to n under certain practical assumptions.

2. Accumulated inner product.

We will assume that the given digital computer will be able to perform the following operations.

(i) Addition and subtraction. The machine accepts numbers in double-precision mantissa and produces a result having a double-precision mantissa.

(ii) Division. The machine accepts a double-precision dividend and a single precision divisor, giving a single-precision quotient.

(iii) Multiplication. The machine accepts single-precision factors and gives a double-precision product.

Furthermore, if a single-precision number has a t-bit mantissa, then a double-precision number will have 2 t-bits for the mantissa. Extending (1.1) to (i), (ii) and (iii), we have the following lemma:

Lemma 2.1. If a, b are single-precision numbers and x, y are double-precision numbers, then we have

$$(1+\delta)\ fl(x+y) = x+y, \quad |\delta| \le 2^{-2t} = u^2; \tag{2.1}$$

$$(1+\delta')\ fl(x-y) = x-y, \quad |\delta| \le 2^{-2t} = u^2; \tag{2.2}$$

$$fl(ab) = ab\ ; \tag{2.3}$$

$$(1+\Delta)\ fl(x/a) = x/a\ , \quad |\Delta| \le 2^{-t} = u\ . \tag{2.4}$$

We see that the results of the operations +, -, and / are the correctly rounded results and the operation multiplication is exact.

We can now consider the computation of the following general inner product by accumulation:

$$p = fl\left[\left(\sum_{i=1}^{n} a_i b_i\right)/y\right] \tag{2.5}$$

The execution of (2.5) can be carried out by the following recursive seouence:

$$\begin{aligned} s_1 &= fl(a_1 b_1), \\ s_{k+1} &= fl(s_k + a_{k+1}b_{k+1}), \qquad 1 \leq k \leq n-1, \\ p &= fl(s_n/y). \end{aligned} \tag{2.6}$$

Applying Lemma 2.1 to (2.6), we have

$$\begin{aligned} s_1 &= a_1 b_1, \\ (1+\delta_{k+1})s_{k+1} &= s_k + a_{k+1}b_{k+1}, \ |\delta_{k+1}| \leq u^2, \ 1 \leq k \leq n-1, \\ (1+\Delta)p &= s_n/y, \qquad |\Delta| \leq u \end{aligned} \tag{2.7}$$

Combining (2.7) for k = 1, 2, ⋯, n-1, we have

$$p + e = \left(\sum_{i=1}^{n} a_i b_i \right) / y \tag{2.8}$$

where

$$e = p\Delta + \frac{1}{y} \sum_{k=1}^{n-1} \delta_{k+1} s_{k+1} \tag{2.9}$$

We note that if $y = 1$ the last step in (2.6) is actually a double-precision to single-precision conversion, hence the last equation in (2.7) is still valid. To bound the error in (2.8), let us denote by σ the magnitude of the absolute maximum of the computed numbers in (2.6), namely,

$$\sigma = \max_{2 \leq k \leq n} (|p|, |s_k|). \tag{2.10}$$

Then the upper bound for e in (2.8) is

$$|e| \leq [1 + \frac{1}{|y|}(n - 1)u]\sigma u \tag{2.11}$$

For $|y| = 1$, then we have a simplified equation

$$|e| \leq [1 + (n - 1)u]\sigma u \tag{2.12}$$

Thus we have established the following lemma:

Lemma 2.2. The accumulated inner product of (2.9) satisfies

$$p + e = \frac{1}{y}\left(\sum_{i=1}^{n} a_i b_i\right) \tag{2.13}$$

where

$$|e| \leq [1 + \frac{1}{|y|}(n - 1)u]\sigma u \tag{2.14}$$

and σ is defined in (2.10).

From (2.14) we see the round-off error in the accumulated inner product is very small if n is not too large and $|y|$ is not too small. This is of course what we have expected in using double-precision additive operations.

3. The Crout algorithm with accumulated inner product.

The usual Gaussian elimination method allows us to obtain the elements of the matrices L and U by a sequence of eliminations of the variables. On the other hand, the Crout algorithm determines the L and U directly from the matrix equation LU = A. Specifically, if L is a unit-diagonal lower triangular matrix, then we have

$$\begin{bmatrix} 1 & & & & & & \\ \ell_{21} & 1 & & & & & \\ \ell_{31} & \ell_{32} & 1 & & & & \\ \cdot & \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & \cdot & & \\ \cdot & \cdot & \cdot & & & \cdot & \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdot & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdot & \cdot & \cdot & u_{1n} \\ & u_{22} & u_{23} & \cdot & \cdot & \cdot & u_{2n} \\ & & u_{33} & \cdot & \cdot & \cdot & u_{3n} \\ & & & \cdot & & & \cdot \\ & & & & \cdot & & \cdot \\ & & & & & \cdot & \cdot \\ & & & & & & u_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdot & \cdot & \cdot & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdot & \cdot & \cdot & a_{3n} \\ \cdot & \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix} \qquad (3.1)$$

If we write out (3.1) in full, we see that the first row of U is given by the equation $u_{11} = a_{11}$, $u_{12} = a_{12}$, $\cdots$, $u_{1n} = a_{1n}$ and the first column of L may then be obtained from the equations $\ell_{21}u_{11} = a_{21}$, $\ell_{31}u_{11} = a_{31}$, $\cdots$, $\ell_{n1}u_{11} = a_{n1}$. We can then solve for the second row of U and the second column of L and so on. The computational equations which give u_{ij} and ℓ_{ij} in terms of previously computed quantities are

$$u_{ij} = fl\left[\left(a_{ij} - \sum_{k=1}^{i-1} \ell_{ik}u_{kj}\right)/1\right] \qquad j \geq i > 1 \tag{3.2}$$

$$\ell_{ji} = fl\left[\left(a_{ji} - \sum_{k=1}^{i-1} \ell_{jk}u_{ki}\right)/u_{jj}\right] \qquad j > i \geq 1 \tag{3.3}$$

Thus the determination of u_{ij} and ℓ_{ji} can be carried out by computing a corresponding accumulated inner product. The partial pivoting strategy can still be employed here if the sequence of (3.2) and (3.3) are slightly altered to allow a search for the largest element in a column as pivot. This is described in Wilkinson [4]. We will assume that the row interchanges has been done in advance so that no pivoting is necessary and the elements of L are all of magnitudes less than or equal to one.

Applying Lemma 2.2 to (3.2) and (3.3), we have

$$u_{ij} + e_{ij} = a_{ij} - \sum_{k=1}^{i-1} \ell_{ik}u_{kj}, \qquad j \geq i > 1, \tag{3.4}$$

$$\ell_{ji} + \varepsilon_{ji} = \frac{1}{u_{ii}}\left(a_{ji} - \sum_{k=1}^{i-1} \ell_{jk}u_{ki}\right), \qquad j > i \geq 1 \tag{3.5}$$

where

$$|e_{ij}| \leq [1 + (i - 1)u]\sigma_{ij}u, \qquad j \geq i > 1 \tag{3.6}$$

and

$$|\varepsilon_{ji}| \leq [1 + \frac{1}{|u_{ii}|}(i - 1)u]\sigma_{ji}u, \qquad j > i \geq 1. \tag{3.7}$$

Now equations (3.4) and (3.5) can also be written as

$$u_{ij} + \sum_{k=1}^{i-1} \ell_{ik}u_{kj} + e_{ij} = a_{ij}, \qquad j \geq i > 1, \tag{3.8}$$

$$\ell_{ji}u_{ii} + \sum_{k=1}^{i-1} \ell_{jk}u_{ki} + \varepsilon_{ji}u_{ii} = a_{ji}, \qquad j > i \geq 1. \tag{3.9}$$

Combining (3.8) and (3.9) in matrix notation, we have

$$LU + F = A \tag{3.10}$$

where $F = (f_{ij})$ and

$$|f_{ij}| = |e_{ij}| \leq [1 + (i - 1)u]\sigma_{ij}u, \qquad j \geq i > 1, \tag{3.11}$$

$$|f_{ji}| = |\varepsilon_{ji}u_{ii}| \leq [|u_{ii}| + (i - 1)u]\sigma_{ji}u, \qquad j > i \geq 1. \tag{3.12}$$

Now let us define

$$\begin{aligned} |F| &= (|f_{ij}|), \\ \rho &= \max_{i}|u_{ii}|, \\ \sigma &= \max_{i,j}|\sigma_{ij}|. \end{aligned} \tag{3.13}$$

Then we have

$$|F| \leq \sigma u \begin{bmatrix} 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \rho & 1+u & 1+u & \cdot & \cdot & \cdot & 1+u \\ \rho & \rho+u & 1+2u & \cdot & \cdot & \cdot & 1+2u \\ \cdot & \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \rho & \rho+u & \rho+2u & \cdot & \cdot & \rho+(n-2)u & 1+(n-1)u \end{bmatrix} \qquad (3.14)$$

The upper bound of the infinite norm of F can thus be estimated as

$$||F||_\infty \leq \begin{cases} [n + \frac{n(n-1)}{2} u]\sigma u, & \text{for } \rho \leq 1; \\ \\ [(n-1)\rho + 1 + \frac{n(n-1)}{2} u]\sigma u, & \text{for } \rho > 1. \end{cases} \qquad (3.15)$$

Note that in (3.15) ρ and σ are not necessarily equal unless column interchanges are done in advance to assure that $\sigma = \rho$.

Thus we have established the following lemma:

Lemma 3.1. The Crout algorithm of directly decomposing A into a product LU by using accumulated inner product gives us triangular matrices L and U such that

$$LU + F = A \qquad (3.16)$$

where the upper bound for F can be estimated using (3.15).

From (3.15) we see that the dominating factor in the error bounds is $n\sigma u$ or $(n-1)\rho\sigma u$ since usually we have $\frac{n(n-1)}{2} u << 1$ for most of the existing general purpose machines. For example, for the IBM 360 series, we have

$t = 24$ and hence $\frac{n(n-1)}{2}\, 2^{-24}$ is approximately equal to 1 if $n \simeq 2900$ which is far more than the system order we encounter in practice. Thus we could ignore this small term and the actual upper bounds for F is approximately proportional to the system order n.

Now we can solve the decomposed system

$$LUx = b \tag{3.17}$$

in the sequence

$$Ly = b \tag{3.18}$$

and

$$Ux = y. \tag{3.19}$$

Again accumulated inner product is used to solve for y and x by the computational equations

$$y_1 = b_1$$

$$y_i = fl\left[\left(b_i - \sum_{j=1}^{i-1} \ell_{ij} y_j\right)/1\right], \qquad 2 \le i \le n \tag{3.20}$$

and

$$x_k = fl\left[\left(y_k - \sum_{j=1}^{k-1} u_{kj} y_j\right)/u_{kk}\right] \qquad 1 \le k \le n \tag{3.21}$$

We can similarly apply Lemma 2.2 to (3.20) and (3.21). The results are summarized in the following lemma:

Lemma 3.2. The computed solutions y and x of the triangular systems (3.18) and (3.19) by the use of accumulated inner product satisfy

$$Ly + e = b \tag{3.22}$$

$$Ux + \varepsilon = y \tag{3.23}$$

where the absolute vectors of e and ε satisfy

$$|e| \leq \sigma_b u \begin{bmatrix} 1 \\ 1+u \\ 1+2u \\ \cdot \\ \cdot \\ \cdot \\ 1+(n-1)u \end{bmatrix} \tag{3.24}$$

$$|\varepsilon| \leq \sigma_y u \begin{bmatrix} \rho+(n-1)u \\ \rho+(n-2)u \\ \cdot \\ \cdot \\ \cdot \\ \rho+(0)u \end{bmatrix} \tag{3.25}$$

and σ_b or σ_y are the magnitude of the absolute maximum value generated during the computation of x or y respectively. Combining Lemma 3.1 and Lemma 3.2, we have the following theorem:

Theorem 3.1. The solution x computed by the Crout algorithm with accumulated inner product satisfies

$$(A + \delta A)x = b + \delta b \tag{3.26}$$

where $\delta A = -F$ and $\delta b = -e - L\varepsilon$. Furthermore,

$$||\delta A||_\infty = ||F||_\infty \tag{3.27}$$

$$||\delta b||_\infty \leq ||e||_\infty + ||L\varepsilon||_\infty \tag{3.28}$$

where

$$||e||_\infty \leq \sigma_b u[1 + (n - 1)u] \tag{3.29}$$

$$||L\varepsilon||_\infty \leq \sigma_y u[n\rho + \frac{n(n-1)}{2} u] \tag{3.30}$$

Thus we see if the assumption that $\frac{n(n-1)}{2} u << 1$ is true, then the computed solution satisfied a perturbed system of (3.26) with upper bounds for the perturbations δA and δb proportional to n. Hence in solving higher order system of linear algebraic equations, the Crout algorithm with accumulated inner product should be used to avoid loss of significance at all stages of computation. This is especially important for ill-conditioned systems where rows or columns are usually more or less dependent.

4. Numerical experiment.

To see how the accumulation of inner product affects the solution accuracy, we have solved a 5 by 5 matrix problem of the type Ax = b where A is a 5-th order inverse Hilbert matrix and $b = (1, 0, 0, 0, 0)^T$. Hence the exact solution is x = (1, 1/2, 1/3, 1/4, 1/5). \n arbitrary precision arithmetic unit [5] is used to simulate a 24-digits mantissa chopped floating-point arithmetic system. The results with and without accumulation of inner products are listed in the following table:

Without Accumulation

x_1 =	0.1000	0000	0000	0000	0000	8947	(10')
x_2 =	0.5000	0000	0000	0000	0006	6930	(10°)
x_3 =	0.3333	3333	3333	3333	3338	6805	(10°)
x_4 =	0.2500	0000	0000	0000	0004	4524	(10°)
x_5 =	0.2000	0000	0000	0000	0003	8143	(10°)

With Accumulation

x_1 =	0.1000	0000	0000	0000	0000	0020	(10')
x_2 =	0.5000	0000	0000	0000	0000	0177	(10°)
x_3 =	0.3333	3333	3333	3333	3333	3488	(10°)
x_4 =	0.2500	0000	0000	0000	0000	0135	(10°)
x_5 =	0.2000	0000	0000	0000	0000	0117	(10°)

Table 4.1. Numerical Results of Solving Ax = b.

We see from Table 4.1 that two more significant digits are obtained in all of the solution components when accumulation of inner products is used in the Crout algorithm. The absolute error in each component is decreased by a factor of 300 to 400 with accumulation.

5. Conclusions.

We have shown, by the a posteriori error analysis, that the computed results of the Crout algorithm with inner product accumulation satisfy a perturbed system and the upper bounds for the perturbations are proportional to the system order n under certain practical assumptions. The improvement in accuracy is basically due to the effort to avoid loss of significance in additive operations. This is confirmed by the results of our numerical experiment. Indeed the inner product accumulation should be done in every computation whenever it is possible.

We should also note that the Crout algorithm is no more than an "analytic" process where first the matrix A is decomposed into factors L and U and later on the vector b is decomposed into L and y and subsequently y is decomposed into U and the desired x. Hence our a posteriori analysis can only give us bounds of the difference between the computed decomposition LU and the exact decomposition A or the difference between the computed decomposition LUx and the exact decomposition b. In order to find the difference between the computed x and the exact solution $A^{-1}b$ we need the information of A^{-1} which is of course unavailable unless the decomposition is also used to obtain an approximate inverse of A.

REFERENCES

[1] Tsao, N. K., Error Analysis of Gaussian Elimination Method for Solving Systems of Linear Algebraic Equations, Aerospace Research Laboratories Technical Report ARL 71-0288, Air Force Systems Command, Wright-Patterson AFB, Ohio, December 1971.

[2] Rice, J. R., A Theory of Condition, SIAM J. of Numer. Anal., 3 (1966), pp. 287-310.

[3] Tsao, N. K., A Posteriori Forward Error Analysis, Aerospace Research Laboratories Technical Report ARL 71-0287, Air Force Systems Command, Wright-Patterson AFB, Ohio, December 1971.

[4] Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice Hall, Englewood, New Jersey, 1963.

[5] Tsao, N. K., and F. F. Kuo, APAU -- An Arbitrary Precision Arithmetic Unit. Proceedings of the Fourth Hawaii International Conference on System Sciences, January 1971, pp. 254-255.